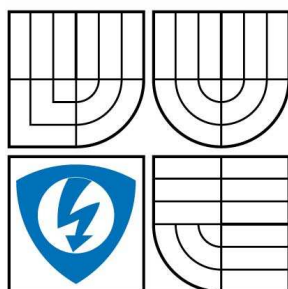


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ**

**FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS**

MONITORING SERVROVÝCH SYSTÉMOV

SERVER SYSTEM MONITORING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARIÁN JANKULÁR

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MIROSLAV BUCHNÍČEK

BRNO 2009

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Marián Jankulár

Bytem: Dolné Vinice 8, Zlaté Moravce, 95301

Narozen/a (datum a místo): 25. marca v Nitre

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 244/53, 602 00, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

prof. Ing. Kamil Vrba, CSc., předseda rady oboru Teleinformatika

(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
- ☐ diplomová práce
- ☐ bakalářská práce
- ☐ jiná práce, jejíž druh je specifikován jako

.....

(dále jen VŠKP nebo dílo)

Název VŠKP:	Monitoring serverových systémů
Vedoucí/ školitel VŠKP:	Ing. Miroslav Buchniček
Ústav:	Ústav telekomunikací
Datum obhajoby VŠKP:	

VŠKP odevzdal autor nabyvateli v * :

- ☐ tištěné formě – počet exemplářů: 1
- ☐ elektronické formě – počet exemplářů: 1

* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ☐ ihned po uzavření této smlouvy
 - ☐ 1 rok po uzavření této smlouvy
 - ☐ 3 roky po uzavření této smlouvy
 - ☐ 5 let po uzavření této smlouvy
 - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel

.....
Autor

ABSTRAKT

Cieľom tejto bakalárskej práce je preštudovanie spôsobu správy a monitoringu serverových systémov. Zmapovanie a popísanie ich základných komponentov a nastavení ako z monitorovacej stránky, tak zo strany monitorovaných systémov a princíp chodu jednotlivých komponentov. Zmapovanie podporovaných operačných systémov a možností komponentov, aplikácií a zdrojov, ktoré je možné monitorovať. Na základe naštudovaných informácií bude nasledovať vytvorenie monitorovacej štruktúry a monitorovaných serverov. Po sprevádzkovaní monitorovacej štruktúry bude nasledovať meranie, z ktorého bude vytvorený zápis

ABSTRACT

Object of this bachelor thesis is to study concept of administration and monitoring server systems. Mapping out and describing basic components of monitoring tools, its options of adjustment from monitoring point of view and also from server adjustment point of view. Mapping out supported operating systems and possibility of component, applications and sources that can be monitored. As a result of this study, there will be created monitoring structure for monitoring servers. After successful creation of the monitoring structure, there will be measurment and report of the measurment provided.

Kľúčové slová

Monitorovanie
Server
Operačný systém
Udalosť
Komponent
Proces
Tivoli Management Region
Tivoli Enterprise Console
Endpoint

Keywords

Monitoring
Server
Operating sytem
Event
Component
Process
Tivoli Management Region
Tivoli Enterprise Console
Endpoint

Prohlášení

Prohlašuji, že svoji Bakalársku prácu na téma Monitoring servrových systémov jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....
podpis autora

Pod'akovanie

Ďakujem vedúcemu semestrálneho projektu Miroslavovi Buchníčkovi za účinnou metódickú, pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovaní mojho semestrálneho projektu.

V Brně dne

.....
podpis autora

Obsah:

Zoznam obrázkov a tabuliek:	7
1. Úvod	8
2. Tivoli Enterprise Console (TEC)	9
2.1. TEC	9
2.2. Komponenty Tivoli Enterprise Console	9
2.2.1. Adapter Configuration Facility	9
2.2.2. Event Adapter	10
2.2.3. Tivoli Event Integration Facility	10
2.2.4. Tivoli Enterprise Console gateway	10
2.2.5. Tivoli NetView	10
2.2.6. Event Server	11
2.2.7. Event databáza	11
2.2.8. User Interface Server (UI server)	11
2.2.9. Event console	12
2.3. Skupiny udalostí	14
2.3.1. Udalosti	16
2.3.2. Tok udalostí	16
2.4. Požiadavky TEC-u	18
3. Inštalácia infraštruktúry	20
3.1. Náhľad na infraštruktúru	20
3.2. Inštalácia Tivoli Management Region serveru	21
3.3. Inštalácia Tivoli Enterprise Console serveru:	24
3.4. Inštalácia Tivoli Endpointu na linuxový systém:	25
3.5. Inštalácia Endpointu a Tivoli Desktop na Microsoft Windows Server 2003 R2 Enterprise Edition:	27
4. Záver	33
5. Zoznam použitých zdrojov	34
6. Zoznam použitých skratiek	35
7. Prílohy	36

Zoznam obrázkov a tabuliek:

- Obr. 2.1: TEC konzola
- Obr. 2.2: Príklad štruktúry Tivoli Enterprise Console
- Tab. 2.1: Kritériá, podľa ktorých je možno vytvárať skupiny udalostí
- Tab. 2.2: Zoznam požadovaných autorizácií na operácie v TEC-u
- Tab. 2.3: Požiadavky komponentov na pre platformu UNIX
- Tab. 2.4: Požiadavky komponentov na pre platformu Windows alebo Linux
- Obr. 3.1: Mnou vytvorená serverová infraštruktúra
- Obr. 3.2: Adresárová štruktúra Tivoli Management Region
- Obr. 3.3: Tivoli Desktop pre Linux
- Obr. 3.4: Vytvorenie Tivoli Gatewaye
- Obr. 3.5: Inštalácia Tivoli produktov na teetec00
- Obr. 3.6: Adresárová štruktúra nainštalovaného endpointu na linuxový systém
- Obr. 3.7: Policy regióny
- Obr. 3.8: Linux Policy región
- Obr. 3.9: Posledný riadok vylistovaných distribučných operácií
- Obr. 3.10: Záznam o úspešnej distribučnej operácii na endpoint tee010ux
- Obr. 3.11: Adresárová štruktúra Tivoli endpointu na OS Windows
- Obr. 3.12: Záznam o úspešnej distribučnej operácii na endpoint 2003ep
- Obr. 3.13: Tivoli Desktop pre Windows
- Obr. 3.14: Prihlasovacie okno do TECu
- Obr. 3.15: Tivoli Enterprise konzola pre región teetmr00
- Obr. 3.16: Hlášky z endpointu 2003ep po vypnutí služieb
- Obr. 3.17: Hlášky z endpointu teetec00 po reštartovaní serveru
- Obr. 3.18: Hláška z endpointu tee010ux o nedostatočnom mieste na disku
- Obr. 3.19: Hláška o nedostupnosti serveru 2003ep
- Obr. 3.20: Hláška o dostupnosti serveru 2003ep
- Obr. 3.21: Informačné hlášky z TEC serveru
- Obr. 3.22: výstup wep ls
- Obr. 3.23: výstup wgateway
- Obr. 3.24: výstup wep 2003ep status
- Obr. 3.25: výstup wep 2003ep
- Obr. 3.26: výstup wdmlseng -e tee010ux
- Obr. 3.27: výstup odadmin odlist

1. Úvod

Táto práca pojednáva o problematike správy a monitoringu serverových systémov, kedy existuje mnoho nástrojov na vzdialenú správu a monitorovanie. V dnešnej dobe veľké firmy, ktoré vlastnia mnoho serverových systémov v rozsiahlej intranetovej alebo internetovej sieti, nestíhajú udržiavať túto infraštruktúru v bezproblémovom chode len pomocou lokálnych zamestnancov, ktorí dohliadajú na chod serverov. Preto prenášajú správu na externé firmy, ktoré vlastnia postupy, ktoré zaručujú jednak rýchly servis a správu predchádzajúcu poruchám, ale hlavne centrálny spôsob riadenia celej infraštruktúry, ktorý poskytuje nadhľad na infraštruktúru, čo pri lokálnej správe nie je možné. Ja sa v mojej práci pokúsim vytvoriť monitorovaciu infraštruktúru pomocou Tivoli softwaru, čo bude zahŕňať inštaláciu serverov Tivoli prostredia a jeho komponentov. Ďalej správne nastavenie Tivoli komponentov a následné odskúšanie pri modelovaných situáciách.

2. Tivoli Enterprise Console (TEC)

Produkt Tivoli Enterprise Console je software založený na udalostných manažment aplikáciách, ktoré monitorujú systémy, sieť, databázy a aplikácie aby bola zaručená optimálna dostupnosť IT zdrojov.

2.1. TEC

- je centralizovaný a globálny náhľad na podnikovú IT infraštruktúru
- zbiera, triedi a dokáže automaticky reagovať na bežné udalosti ako napríklad databázový server je nedostupný, sieťový výpadok alebo dokončená operácia
- sa správa ako centrálny zberací bod pre alarmy a udalosti z rôznych zdrojov zahrňujúcich Tivoli aplikácie, Tivoli partnerské aplikácie, vlastné aplikácie, sieťové platformy pre manažment a databázové systémy
- obsahuje Tivoli NetView komponent – jednotný systém a sieťový manažment

TEC pomáha efektívne spracovávať veľké množstvo udalostí v IT prostredí pomocou:

- triedením udalostí podľa stupňa dôležitosti
- odfiltrávaním prebytočných udalostí a udalostí s nízkou prioritou
- zlad'ovať udalosti s inými udalosťami z iných zdrojov
- rozhodovania, kto by mal danú udalosť vidieť
- zahájením automatickej akcie ako napríklad postúpením hlášky, vytvorením tiketu (dokumentácia riešenia daného problému)
- identifikovaním systému, ktorý je práve v procese údržby a zbieraním hlášiek z tohto systému
-

Jednotný systém a sieťový manažment poskytuje:

- zistenie, či príčina nedostupnosti systému alebo aplikácie nie je sieťový problém
- ladenie statusu systémov alebo aplikácií so statusom siete, ktorú systém/aplikácia používa
- automatické spracovanie sieťových udalostí, zobrazovanie len tých udalostí, na ktoré je potrebný operátor
- možnosť predefinovaných skupín ale aj možnosť vytvorenia vlastných skupín založených na určitých pravidlách (napr: geografická poloha, druh hlášiek, dôležitosť systému ...)

2.2. Komponenty Tivoli Enterprise Console

- Adapter Configuration Facility
- Event Adapter
- Tivoli Event Integration Facility
- Tivoli Enterprise Console Gateway
- Tivoli NetView
- Event Server
- Event database
- User interface server
- Event console

2.2.1. Adapter Configuration Facility

Konfiguračné prostriedok pre adaptéry poskytuje grafické užívateľské prostredie pre konfiguráciu a distribúciu TME (Tivoli Management Environment) adaptérov. Adapter Configuration Facility môže byť použitá na vytváranie profilov pre adaptéry, nastavenie konfigurácie adaptérov a možnosti ich distribúcie. Toto umožňuje vykonávať zmeny

centrálne a distribúciu nastavení do vzdialených lokácií ku koncovým užívateľom alebo spravovaným uzlom.

2.2.2. Event Adapter

Adaptér udalostí je proces bežiaci na Endpointe, ktorý typicky prebýva na tom istom spravovanom zdroji a monitoruje zdrojové udalosti. Zdroj je aplikácia (napr: databáza) alebo systémový prostriedok (napr: voľný priestor na disku). Ak chceme monitorovať napr. logy na systémoch, na ktorých je nainštalovaný MS Windows Server 2003 musí byť na každom tomto systéme bežať Event Adapter alebo musí byť systém modifikovaný tak aby odosielať informácie do TEC –u. Adaptér môže byť nakonfigurovaný, že bude sám zahadzovať prebytočné hlášky a tým bude redukovať zaťaženie siete. Event Adapter môže byť nastavený tak, že bude odosielať informácie do Event serveru pomocou Tivoli rozhrania – využíva služby a prostriedky Tivoli Management Framework alebo non-Tivoli rozhrania kedy môžu byť informácie zasielané otvorením IP socketu. Adaptér sa pripája k Event Serveru priamo alebo pomocou Tivoli Event Console gateway. Event Adapter môže ukladať udalosti do vyrovnávacej pamäti, táto možnosť je výhodná a zapína sa v prípade, že sa adaptér nemôže pripojiť k Event Serveru. Akonáhle je pripojenie uskutočnené adaptér odošle všetky uložené udalosti do Event Serveru kde sú následne spracované. Adaptéry môžu byť nakonfigurované tak že v prípade výpadku primárneho Event Serveru budú odosielať informácie do sekundárneho Event Serveru. Adaptér odosiela hlášky do endpoint procesu lcfd a ten pomocou Tivoli rozhrania komunikuje s Tivoli bránou alebo samotným Event serverom.

2.2.3. Tivoli Event Integration Facility

Integračné prostredie pre udalosti je nástroj, ktorý umožňuje rozšíriť typy udalostí a systémové informácie, ktoré môžu byť monitorované, taktiež umožňuje vytvorenie vlastných adaptérov, ktoré sú prispôbené na monitorovanie špeciálnych udalostí.

2.2.4. Tivoli Enterprise Console gateway

Tivoli brána prijíma informácie z TME aj non-TME adaptérov a preposiela ich do Event Serveru. Tivoli brána komunikuje s Event serverom pomocou Tivoli Event Integration Facility. Tivoli brána je zložená z tec_gateway a tec_gwr programov. Tec_gateway program spracováva informácie z TME adaptérov. TEC_gateway a všetky ostatné potrebné súbory pre adaptéry pre každý koncový operačný systém sú nainštalované, keď je nainštalovaný Adapter Configuration Facility na spravovanom uzle, ktorý je nakonfigurovaný ako Tivoli Enterprise Console gateway.

Tivoli brána prináša tieto benefity:

- väčšiu dostupnosť, ktorá umožňuje spravovať zdroje s minimom softwaru bežiacich na koncových zariadeniach
- zlepšuje chod Event Serveru tým, že zahadzuje a zoskupuje informácie do neho odosielané do skupín podľa podobnosti informácie a tým uľahčuje jeho chod
- jednoduché rozmiestnenie adaptérov a updatov na adaptéry použitím profilov v Adapter Configuration Facility

2.2.5. Tivoli NetView

Tento komponent zabezpečuje správu funkčnosti sieťových prvkov, ktoré využíva TEC. Tivoli NetView monitoruje tieto zariadenia a v prípade výpadku jedného z nich automaticky filtruje chybné hlášky z adaptérov za týmito zariadeniami. Ak by napríklad došlo k výpadku jednej z Tivoli brán mohlo by dochádzať k zahlcovaniu Event Serveru,

konzoly pre operátorov hláškami o nedostupnosti jednotlivých systémov, aplikácií atď. a pritom by dostupné boli, nedostupné sú len pre monitoring. V tomto prípade Tivoli NetView označí danú bránu za nedostupnú a hlášky, ktoré sa mali hlásiť z nej do Event Servru budú ignorované po tú dobu až bude táto brána zase dostupná.

2.2.6. Event Server

Tento server poskytuje centrálnu správu udalostí v distribuovanom prostredí. Každý Tivoli región obsahuje len 1 Event Server. Tento server spracováva vstup z konzoly udalostí a aktualizuje databázu udalostí. Pretože konzola udalostí načítava dáta z databázy udalostí pre každú prichádzajúcu udalosť a potom zhodnocuje tieto udalosti s určitými pravidlami aby zistil, či má Event Server automaticky vykonať nejakú preddefinovanú úlohu alebo upraviť udalosť a odoslať ju operátorovi aby vykonal operáciu na odstránenie problému. Ak Event Server prijme udalosť, ktorá je dôležitá, tak jej prideli jedinečný identifikátor a časovú známku a prepošle ho do vyhodnocovacieho nástroja (Rule engine process) pre ďalšie spracovanie. Ak je vyhodnocovací nástroj zamestnaný a nemôže udalosť prijať, Event Server uloží udalosť do bufferu pokým ho nebude môcť vyhodnocovací nástroj spracovať. Spracované udalosti sú uschované v databáze udalostí. Event Server reaguje na každú príchodziu udalosť tým, že zisťuje či má na danú udalosť vytvorené nejaké pravidlo, podľa ktorého má s udalosťou zachádzať. Pravidlo obsahuje sadu výrazov užívaných na porovnanie udalosti a podmienok pravidla. Tieto pravidlá pomáhajú taktiež redukovať nadbytočné hlášky odosielané operátorom. Event Server môže taktiež obsahovať pravidlá na oddialenie odoslania hlášky operátorovi, takéto pravidlá sa uplatňujú hlavne na problémy, ktoré sa dokážu vyriešiť na systéme samy bez akéhokoľvek zásahu. (napr. spadnutie automatickej služby, ktorá sa sama po čase naštartuje) Pravidlá špecifikujú ale nie sú obmedzené nasledujúcimi akciami:

- zľadovanie udalostí
- automatické reagovanie na niektoré udalosti
- oddľalovanie reakcie na udalosť
- pozmeňovanie atribútov udalostí
- predchádzanie odosielanie duplikátnych hlášiek operátorom
- expedovanie Tivoli a iných administratívnych akcií na zdroje
- prehodnocovanie sád udalostí
- zahadzovanie udalostí
- generovanie nových udalostí
- presmerovávanie udalostí na iný Event Server

2.2.7. Event databáza

TEC používa externú relačnú databázu správy systému (RDBMS – relational database management system) na ukladanie veľkého množstva dát udalostí, ktoré prichádzajú z Event Serveru. Na prístup do RDBMS databázy používa RDBMS Interface Module (RIM).

2.2.8. User Interface Server (UI server)

Tento server zabezpečuje komunikáciu medzi konzolami a event serverom. UI server komunikuje s expedičným nástrojom (Dispatch engine process) kedy má kontaktovať event server. UI server zabezpečuje aby neboli konzoly zahlcované hláškami o tom istom probléme. UI server taktiež aktualizuje status hlášky na každej konzole, tak že odošle zmenu stavu do expedičného nástroja, ktorý odošle zmenu do databázy udalostí kde je tento stav uschovaný. Napríklad jeden z operátorov vyrieši problém a tak zavrie

hlášku v konzole a UI server zabezpečí aby sa na ďalších konzoliach táto hláška tak isto zavrela. UI server taktiež poskytuje sadu príkazov, ktoré umožňujú operátorovi vlastnosť udalosti, vylistovať udalosti z určitej skupiny alebo zobrazenie konkrétnej udalosti. UI server môže byť nainštalovaný na akomkoľvek spravovanom uzle v 1 Tivoli regióne, ale 1 Tivoli región môže obsahovať len 1 UI server. UI server nemusí byť nainštalovaný na tom istom systéme ako je event server. Proces UI serveru sa nazýva `tec_ui_server`.

2.2.9. Event console

Konzola udalostí poskytuje grafický užívateľské rozhranie (GUI), v ktorom sa dá ľahko orientovať a uľahčuje operátorovi reagovať na problémy v infraštruktúre. TEC poskytuje 2 druhy konzol – Java a Web verziu. Administrátor musí používať Java verziu konzoly na konfiguráciu konzoly. Java verzia je taktiež nevyhnutná pre použitie Tivoli NetView funkcií alebo vykonávanie lokálnych automatizovaných úloh. Pre spravovanie je vhodná ako Web verzia tak aj Java verzia.

2.2.9.1. Java verzia

Môže byť nainštalovaný na spravovanom uzle, koncovom systéme alebo aj počítači v non-Tivoli prostredí a zahŕňa nasledujúce možnosti:

- Tivoli bezpečné prihlásenie
- každá konzola prijíma informácie priamo z databázy
- možnosť nastavenia doby aktualizácie konzoly z databázy
- možnosť vykonávať lokálne akcie na vzdialených systémoch, taktiež možnosť používania skriptov alebo aplikácií
- možnosť vykonávať preddefinované úlohy
- možnosť nastavenia prevedenia určitej akcie pri prijatí danej hlášky v konzole
- možnosť vidieť podrobnejšie informácie na webovej stránke po otvorení problému v konzole, čo pomôže operátorovi pri riešení problému alebo kontaktovaní zodpovednej osoby
- možnosť zoskupovať viac udalostí z jedného zdroja, čo tak isto pomôže operátorovi detailnejšie pochopiť problém
- viac operátorov môže dohliadať na jednu konzolu
- editáciu definícií konzoly získajú okamžite po zmene všetci prítomní operátori
- možnosť importu a exportu nastavení konzoly pre ľahšiu migráciu z developérskeho prostredia do produkčného prostredia

Java verzia poskytuje nasledovné zobrazenia:

- 1, Konfiguračné zobrazenie: - využíva sa konfiguráciu Java a Web verzie, len administrátori majú prístup do tohto zobrazenia
- 2, Zhrňujúce grafické zobrazenie: - poskytuje pohľad na rozdelenie skupín, ktoré sú reprezentované stĺpcovými grafmi, ktoré odrážajú momentálny stav a počet udalostí v skupine
- 3, Prioritné zobrazenie: - skupiny sú reprezentované tlačidlami, ktoré sú sfarbené do farby aký najvyšší level hlášky skupina obsahuje

2.2.9.2. Web verzia

Zahŕňa nasledujúce možnosti:

- Tivoli bezpečné prihlásenie
- možnosť vidieť podrobnejšie informácie na webovej stránke po otvorení problému v konzole, čo pomôže operátorovi pri riešení problému alebo kontaktovaní zodpovednej osoby
- možnosť zoskupovať viac udalostí z jedného zdroja, čo tak isto pomôže operátorovi detailnejšie pochopiť problém
- viac operátorov môže dohliadať na jednu konzolu
- zmeny definícií konzoly vykonané v Java verzii sa prejavia vo Web verzii pri nasledovnom prihlásení
- možnosť nastavenia doby aktualizácie konzoly z databázy
- možnosť vykonávať preddefinované úlohy

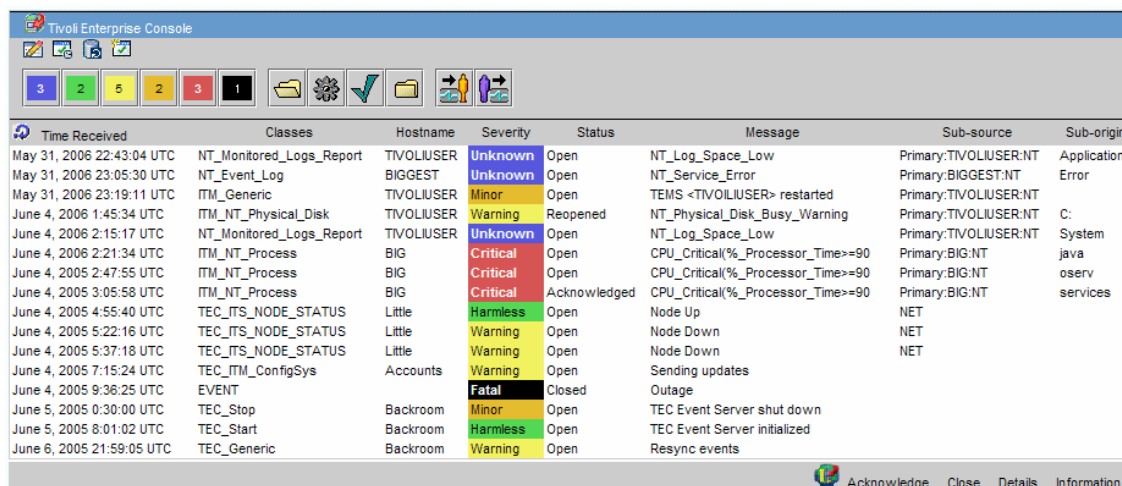
Web verzia organizuje úlohy do portfólia, ktoré je nazvané „My Work“. Portfólio obsahuje nasledujúce operácie:

1, Výber skupiny udalostí: - zoznam skupín udalostí, ktoré boli pridelené do konzoly

2, Zhrňujúce grafické zobrazenie: - poskytuje pohľad na rozdelenie skupín, ktoré sú reprezentované stĺpcovými grafmi, ktoré odrážajú momentálny stav a počet udalostí v skupine

3, Vykonávať úlohy: - vykonáva preddefinovaných úloh, ktoré sa načítavajú z knižníc úloh

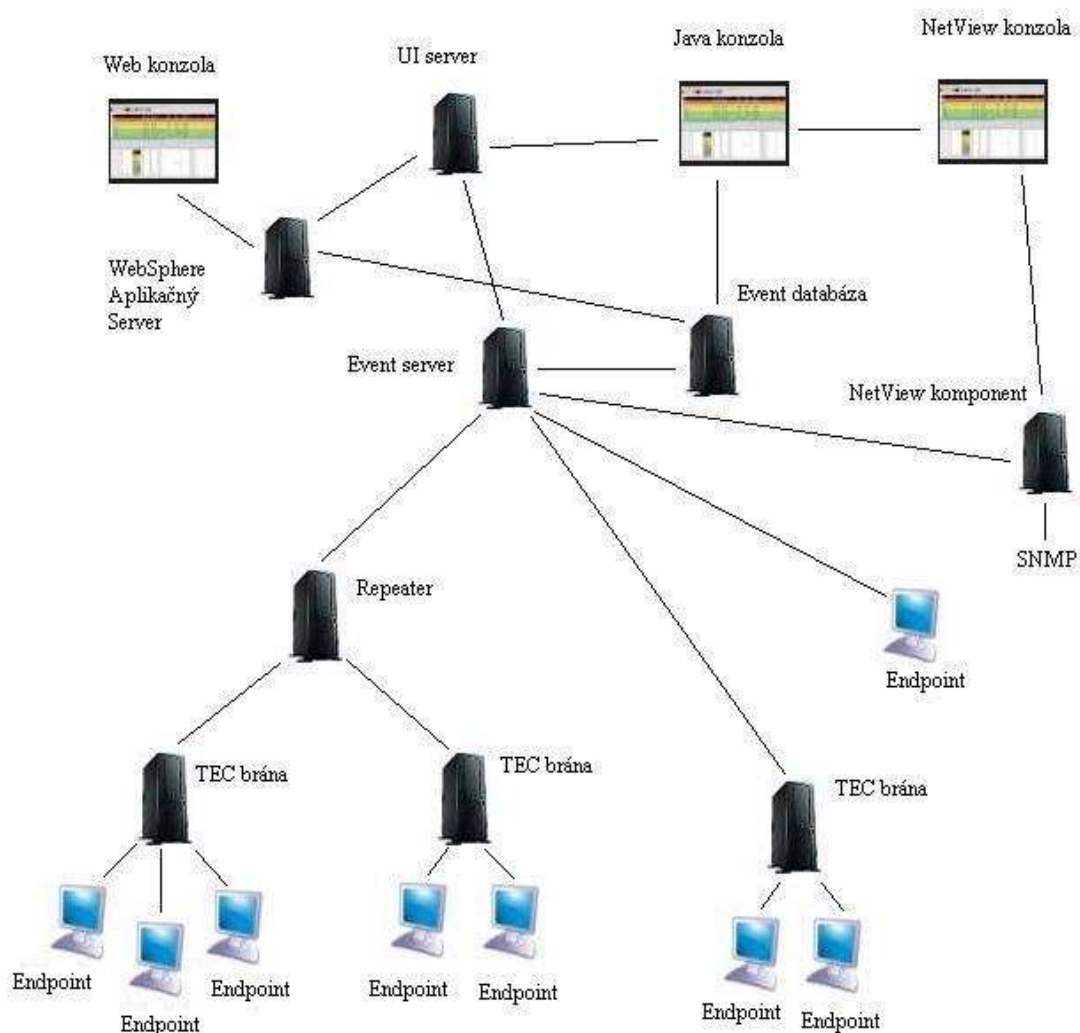
4, Zmena užívateľských nastavení: - poskytuje kontrolu nad zobrazovacími možnosťami ako napríklad doba aktualizácie udalostí, alebo počet udalostí zobrazovaných v konzole.



The screenshot shows the Tivoli Enterprise Console interface. At the top, there's a title bar 'Tivoli Enterprise Console' and a toolbar with various icons. Below the toolbar is a table of events. The table has columns: Time Received, Classes, Hostname, Severity, Status, Message, Sub-source, and Sub-origin. The events are listed with their respective details, including timestamps, class names, hostnames, severity levels (Unknown, Minor, Critical, Harmless, Fatal), and status (Open, Reopened, Acknowledged, Closed).

Time Received	Classes	Hostname	Severity	Status	Message	Sub-source	Sub-origin
May 31, 2006 22:43:04 UTC	NT_Monitored_Logs_Report	TIVOLIUSER	Unknown	Open	NT_Log_Space_Low	Primary:TIVOLIUSER:NT	Application
May 31, 2006 23:05:30 UTC	NT_Event_Log	BIGGEST	Unknown	Open	NT_Service_Error	Primary:BIGGEST:NT	Error
May 31, 2006 23:19:11 UTC	ITM_Generic	TIVOLIUSER	Minor	Open	TEMS <TIVOLIUSER> restarted	Primary:TIVOLIUSER:NT	
June 4, 2006 1:45:34 UTC	ITM_NT_Physical_Disk	TIVOLIUSER	Warning	Reopened	NT_Physical_Disk_Busy_Warning	Primary:TIVOLIUSER:NT	C:
June 4, 2006 2:15:17 UTC	NT_Monitored_Logs_Report	TIVOLIUSER	Unknown	Open	NT_Log_Space_Low	Primary:TIVOLIUSER:NT	System
June 4, 2006 2:21:34 UTC	ITM_NT_Process	BIG	Critical	Open	CPU_Critical(%_Processor_Time)>=90	Primary:BIG:NT	java
June 4, 2006 2:47:55 UTC	ITM_NT_Process	BIG	Critical	Open	CPU_Critical(%_Processor_Time)>=90	Primary:BIG:NT	oserv
June 4, 2006 3:05:58 UTC	ITM_NT_Process	BIG	Critical	Acknowledged	CPU_Critical(%_Processor_Time)>=90	Primary:BIG:NT	services
June 4, 2006 4:55:40 UTC	TEC_ITS_NODE_STATUS	Little	Harmless	Open	Node Up	NET	
June 4, 2006 5:22:16 UTC	TEC_ITS_NODE_STATUS	Little	Warning	Open	Node Down	NET	
June 4, 2006 5:37:18 UTC	TEC_ITS_NODE_STATUS	Little	Warning	Open	Node Down	NET	
June 4, 2006 7:15:24 UTC	TEC_ITM_ConfigSys	Accounts	Warning	Open	Sending updates		
June 4, 2006 9:36:25 UTC	EVENT		Fatal	Closed	Outage		
June 5, 2006 0:30:00 UTC	TEC_Stop	Backroom	Minor	Open	TEC Event Server shut down		
June 5, 2006 8:01:02 UTC	TEC_Start	Backroom	Harmless	Open	TEC Event Server initialized		
June 6, 2006 21:59:05 UTC	TEC_Generic	Backroom	Warning	Open	Resync events		

Obr. 2.1: TEC konzola



Obr. 2.2: Príklad štruktúry Tivoli Enterprise Console

2.3. Skupiny udalostí

Skupiny udalostí je nastavená logická oblasť, ktorá spĺňa určité sadu kritérií. Skupiny sú konfigurované administrátorom v Java verzii. Na definovanie skupine je potrebné špecifikovať kritériá pre udalosti v skupine. Tieto kritériá ustanovuje „event group filter“. Tento filter zahŕňa akýkoľvek znak udalostí s výnimkou podrobného alebo zákazníkom definovaného znaku.

Trieda udalosti	Špecifikuje triedu udalosti podľa zdroja, ktorý odoslal udalosť. Dokumentácia pre adaptér vylistuje možné hodnoty triedy pre adaptér. A ten odošle udalosť s príslušnou triedou.
Pôvod	Identifikuje protokolovú adresu alebo meno hosta z ktorého chceme prijímať udalosti. Ak chceme monitorovať celú subdoménu vložíme protokolovú adresu alebo meno subdomény.
Severita	TEC používa nasledujúce severity: UNKNOWN HARMLESS WARNING MINOR CRITICAL FATAL
Zdroj	Špecifikuje typ aplikácie, ktorá vytvorila udalosť. Zdroje sú definované v adaptéry alebo dokumentácii aplikácie. Možné hodnoty zahŕňajú ale nelimitujú LOGFILE, NT, HO OpenView a SNMP.
Status	ACKNOWLEDGED – operátor alebo pravidlo berie udalosť na vedomie CLOSED – operátor alebo pravidlo zavrelo udalosť pretože problém bol odstránený OPEN – pretrvávajúci problém, ktorý operátor ešte nevyriešil Organizačne dodaný status – numerický status, ktorý je poslaný do TEC – u a zahrňuje rezervované miesto pre status, ktorý môže systémový administrátor nastaviť na požiadanie RESPONSE – Pravidlo automaticky reagovalo na udalosť.

Tab. 2.1: Kritériá, podľa ktorých je možno vytvárať skupiny udalostí

Je možné použiť SQL wildcard pre akýkoľvek atribút na zjednodušenie filtrovania do skupín udalostí. Zadaním špeciálnej hodnoty pre každý atribút sa zníži typ udalostí prichádzajúcich do skupiny. Použitím SQL wildcard je možné rozšíriť počet typov prichádzajúcich do skupiny.

TEC vyžaduje pre vlastné nastavovanie určitú úroveň administrátorských práv.

V nasledujúcej tabuľke je uvedený ich prehľad:

Aktivita	Objekt	Tivoli autorizácia
Pridávanie administrátorských práv	Server udalostí	senior
Konfigurácia		super
Zapínanie a vypínanie		senior
Pridávanie operátora do konzoly	Konzola udalostí	senior
Konfigurácia vlastných nastavení prehliadania		user
Pridelovanie práv v skupine	Skupiny udalostí	senior
Vytváranie, editácia a mazanie		senior
Pridelovanie do skupiny		senior
Mazanie	Udalosti	senior
Zatváranie a skrývanie		admin, RIM_view, RIM_update
Prezeranie		user a RIM_view
Zasielanie		user ak používa príkaz "wpostmsg"
Vytváranie, editácia a načítavanie	Základňa pravidiel	senior
Inštalácia TEC	Tivoli región	super

Tab. 2.2: Zoznam požadovaných autorizácií na operácie v TEC-u

2.3.1. Udalosti

V TEC-u je udalosť objekt, ktorý bol vytvorený na základe nejakých dát alebo udalostí na adaptérom monitorovanom systéme. Každá udalosť je identifikovaná triedou, ktorý adaptér identifikuje. Triedy udalostí rozdeľujú udalosti, ale udalosti obsahujú aj dodatočné informácie. Pre každú triedu sú v TEC-u pripravené predlohy pre udalostné hlášky, ktoré sú modifikovateľné tak, aby poskytovali čo najviac informácií.

Na posielanie udalostí z adaptérov na server je dostupných niekoľko komunikačných protokolov. Udalosti využívajú vernostných atribútov pre potvrdenie, že udalosť bola prijatá prvkom Tivoli infraštruktúry. Vernostná hodnota 1 poukazuje na udalosť prišlú z TME adaptéru, vernostná hodnota 0 naopak z non-TME adaptéru. Komunikácia medzi event serverom a konzolami je na tej istej bezpečnostnej úrovni ako ostatná komunikácia v Tivoli infraštruktúre. Udalosti sú odosielané ako holý text, ktorý nie je šifrovaný. TEC generuje správny chod interné správy o svojom chode. Existuje 5 typov interných správ:

- 1, TEC_Notice – správa, ktorá obsahuje bez-errorovú hlášku o podmienkach v TEC-u
- 2, TEC_Error – správa, ktorá obsahuje správu o konkrétnom probléme v TEC-u
- 3, TEC_DB – správa, ktorá hovorí zlyhaní prístupu do databázy
- 4, TEC_Start – správa vygenerovaná pri štarte Event serveru
- 5, TEC_Stop – správa vygenerovaná pri stopnutí Event serveru

2.3.2. Tok udalostí

Architektúra Event serveru je zložená z nasledujúcich 5 procesov:

- 1, Master proces – tec_server
- 2, Reception engine proces – tec_reception
- 3, Rule engine proces – tec_rule
- 4, Dispatch engine proces – tec_dispatch
- 5, Task engine proces – tec_task

2.3.2.1. Master proces

Tento proces koordinuje všetky ostatné procesy na Event servery. Ak jeden z procesov zastaví alebo zlyhá, master proces sa ho pokúsi reštartnúť. Ak sa mu nepodarí tento proces reštartnúť, tak za staví všetky ostatné procesy na Event servery. V tomto momente je vydaná správa ktorá je odoslaná do recepčného procesu.

2.3.2.2. Reception engine proces

Tento proces prijíma udalosti a ukladá ich do logov, taktiež odosiela udalosti do Rule engine procesu pre ďalšie spracovanie. V procese prijímania udalosti prijme tento proces udalosť priamo z non-Tivoli komunikácie alebo z Tivoli komunikácie z Master procesu. Všetky udalosti sú zapísané do recepčného logu, čo je vlastne tabuľka v databáze udalostí a sú označené ako QUEUED alebo WAITING. Tie, ktoré sú označené ako QUEUED sú poslané do recepčnej pamäte, z tejto pamäte sú odoslané do Rule engine procesu na spracovanie. Expedičný proces signalizuje recepčnému procesu hneď ako dokončí spracovávanie nejakej udalosti, recepčný proces označí udalosť ako PROCESSED v recepčnom logu a vymaže ju z recepčnej pamäte. Ak Rule engine proces nemôže nejakú udalosť spracovať, tak táto udalosť je označená ako PARSING_FAILED v recepčnom logu.

QUEUED – udalosť čaká v recepčnej pamäti na odoslanie do Rule engine procesu. Táto pamäť je prvé miesto kde sa udalosť ukladá a prvé miesto odkiaľ odchádza nejak spracovaná. Príkaz „wtdump“ nám pomáha doladiť veľkosť pamäte a skontrolovať, či Rule engine proces pracuje efektívne. Ak je väčšina udalostí po dokončení tohto príkazu

označená ako QUEUED, Rule engine proces je príliš zaneprázdnený, ak je väčšina udalostí označená ako PROCESSED pamäť má správnu veľkosť a Rule engine proces efektívne vykonáva svoju úlohu.

WAITING – udalosti sú takto označené ak je recepčná pamäť plná. Ak je po vykonaní príkazu „wtdump“ väčšina udalostí označená WARNING pamäť je príliš malá alebo Rule engine pracuje neefektívne

PROCESSED – Takto označená udalosť je spracovaná a platná.

PARSING_FAILED - táto udalosť je neplatná. Buď Event server nevie rozoznať triedu udalosti alebo atribúty triedy nie sú v správnom formáte. Táto udalosť je zahodená, to znamená, že recepčný proces nevloží udalosť do uložiska udalostí ale udalosť je možné vidieť len v recepčnom logu.

Po reštarte Event serveru sa načítavajú do recepčného procesu udalosti z recepčného logu označené ako WAITING a QUEUED. Recepčný proces nespracováva interne generované udalosti – nenájde ich v recepčnom logu ani pamäti. Recepčná pamäť je lokalizovaná v pamäti RAM a nastavovať ju môžeme pomocou príkazu „wsertesvcfg“ z parametrového dialógu Event serveru.

2.3.2.3. Rule engine proces

Tento proces spracováva udalosti na základe vopred zadanych pravidiel. Rule engine proces porovnáva každú jednu udalosť oproti sade pravidiel a počas tejto doby má prívlastok – analyzovaný. Ak udalosť prejde kritériami pravidla, spôsobí to zadefinovanú akciu pravidla, to je prvý vstup do spracovania udalosti. Druhý vstup prichádza zo strany expedičného procesu, tento vstup vykoná zmeny na udalostiach, ktoré boli prijaté v Event servere v minulosti podľa toho aká aktivita bola s nimi vykonaná v konzole. Tretí vstup je zo strany task procesu cez expedičný proces, ak má task proces pravidlo na riešenie udalosti. Po týchto akciách Rule engine proces hodnotí a zladuje udalosti v pamäti.

2.3.2.4. Dispatch engine proces

Hlavná úloha expedičného procesu je udržiavať databázu aktualizovanú a posielat aktualizované hlášky do konzol. Expedičný proces komunikuje s Rule a Task engine procesmi by vedel kedy má aktualizovať databázu udalostí. Ak je nutné spustiť nejaký program alebo skript tak expedičný proces inicializuje akciu Task engine procesu. Expedičný proces tak isto aktualizuje databázu udalostí podľa aktivity v konzole. Ak je Event server reštartovaný, tak expedičný proces načíta udalosti z databázy aby pamäť Rule engine procesu neobsahovala tie isté udalosti.

2.3.2.5. Task engine proces

Proces ktorý spúšťa skripty, úlohy, príkazy a aplikácie vyvolané podľa pravidiel. Počas celej akcie ich monitoruje a po skončení odošle status do expedičného procesu, ktorý zapíše status do databázy udalostí. Task engine proces zahajuje akcie podľa toho ako dostáva požiadavky, nečaká na dokončenie akcie pred tým ako začne inú akciu.

2.4. Požiadavky TEC-u

Platforma	Komponent	Odporúčaný HW	Minimálny HW	Odporúčaná RAM	Minimálna RAM	Miesto na disku
UNIX	Event server + event databáza + RIM na 1 servry	2 procesory - SPECint2000 aspoň 10	1 procesor - SPECint2000 aspoň 5	1GB	512MB	512MB
	Event server + event databáza + RIM + WebSphere aplikačný server na 1 servry	4 procesory - SPECint2000 aspoň 30	2 procesory - SPECint2000 aspoň 10	4GB	2GB	8GB
	WebSphere aplikačný server	1 procesor - SPECint2000 aspoň 10	1 procesor - SPECint2000 aspoň 5	2GB	2GB	2GB
	UI server	1 procesor - SPECint2000 aspoň 150	1 procesor - SPECint2000 aspoň 90	512MB	256MB	256MB
	TEC gateway	1 procesor - SPECint2000 aspoň 150	1 procesor - SPECint2000 aspoň 90	512MB	256MB	256MB
	Event konzola	1 procesor - SPECint2000 aspoň 90	1 procesor - SPECint95 aspoň 7	256MB	128MB	64MB

Tab. 2.3: Požiadavky komponentov na pre platformu UNIX

Platforma	Komponent	Odporúčaný HW	Minimálny HW	Odporúčaná RAM	Minimálna RAM	Miesto na disku
Windows/Linux	Event server + event databáza + RIM na 1 servry	2 procesory - 1.8 GHz P4	1 procesor - 1.8 GHz P4	1GB	512MB	512MB
	Event server + event databáza + RIM + WebSphere aplikačný server na 1 servry	4 procesory - 1.8 GHz P4	2 procesory - 1.8 GHz P4	4GB	2GB	8GB
	WebSphere aplikačný server	1 procesor - 1.8 GHz P4	1 procesor - 933 MHz P3	2GB	2GB	2GB
	UI server	1 procesor - 933 MHz P3	1 procesor - 733 MHz P3	512MB	256MB	256MB
	TEC gateway	1 procesor - 933 MHz P3	1 procesor - 733 MHz P3	512MB	256MB	256MB
	Event konzola	1 procesor - 933 MHz P3	1 procesor - 450 MHz P3	256MB	128MB	64MB

Tab. 2.4: Požiadavky komponentov na pre platformu Windows alebo Linux

Event server, NetView, Adapter Configuration Facility, UI server, Java verziu konzoly je možné nainštalovať na:

- IBM AIX
- Solaris
- HP-UX (s výnimkou NetView)
- Windows Server 2000, 2003
- IBM zOS
- SuSE Linux Enterprise Server zSeries/IA32
- United Linux zSeries (s výnimkou konzoly)
- Redhat Linux zSeries (s výnimkou konzoly)
- Redhat Linux Advanced server IA32

Adaptéry je možné nainštalovať na nasledujúce OS:

- IBM AIX
- Sun Solaris Operating Environment
- HP-UX
- Windows XP Pro
- Windows Server 2000, 2003, 2008
- Windows Datacenter 2000
- IBM zOS
- SuSE Linux Enterprise Server IA32
- SuSE Linux Enterprise Server iSeries
- SuSE Linux Enterprise Server zSeries
- SuSE Linux Enterprise Server pSeries
- RedHat Linux Enterprise Server iSeries
- Redhat Linux Enterprise Server zSeries
- Redhat Linux Enterprise Server pSeries
- Netware
- OS/2
- OS/400
- OS/2 Server eBus
- Compaq Tru64
- Reliant UNIX
- SCO Unixware
- Irix SGI
- Solaris ix86
- Sequent Dynix/PTX

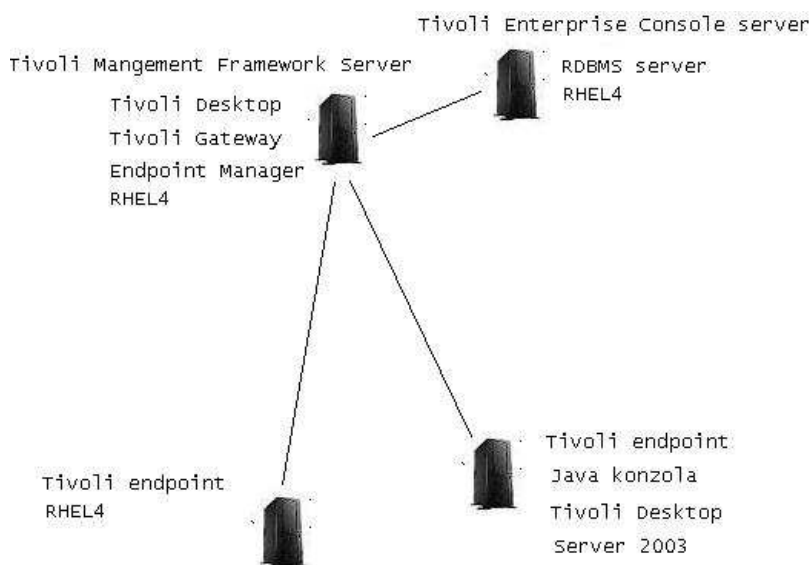
3. Inštalácia infraštruktúry

3.1. Náhl'ad na infraštruktúru

Môj prvotný plán bol nainštalovať 10 virtuálnych serverov z toho 5 bežiacich s OS Microsoft Windows Server 2003 R2 Enterprise Edition a 5 s OS Red Hat Linux Enterprise 4. Bohužiaľ nevlastním výkonný počítač aby bol schopný utiahnuť požiadavky mnou plánovanej infraštruktúry. Preto som po odporúčení môjho vedúceho projektu pána Miroslava Buchníčka zvolil koncept 3 OS bežiacich na OS Red Hat Linux Enterprise 4 a 1 server bežiaci na Microsoft Windows Server 2003 R2 Enterprise Edition, pričom na 1 linuxovom servery bude spustený Tivoli Management Framework, Tivoli Gateway, Tivoli Desktop a Endpoint Manager. Na druhom linuxovom servery bude nainštalovaný Tivoli Enterprise Console server, RDBMS server. Na poslednom linuxovom servery bude nainštalovaný Tivoli endpoint a tento server poslúži len ako monitorovaný server v sieti. Na servery s OS Windows bude nainštalovaný Tivoli endpoint , takže tento server sa bude taktiež monitorovať. Navyiac na tomto servery bude nainštalovaný Tivoli Desktop pre OS Windows. Na emuláciu serverových OS som použil 30 dňovú trial verziú VMware Workstation 6.5.2.

Postupne som nainštaloval Red Hat Enterprise Linux 4 pre Tivoli Management Region server, Tivoli Enterprise Console server a 2 infraštruktúrne servery, jeden s Linuxovou distribúciou Red Hat Enterprise Linux 4 a druhý Microsoft Windows Server 2003 R2 Enterprise. Operačné systémy zastupujú najrozšírenejšiu skupinu používaných OS vo firemných štruktúrach. Inštalácia Tivoli infraštruktúry bude prebiehať postupne v tomto poradí:

- 1, Tivoli Management Region server - Red Hat Enterprise Linux 4
- 2, Tivoli Enterprise Console server - Red Hat Enterprise Linux 4
- 3, Endpoint - Red Hat Enterprise Linux 4
- 4, Endpoint – Windows Server 2003 R2 Enterprise Edition



Obr. 3.1: Mnou vytvorená serverová infraštruktúra

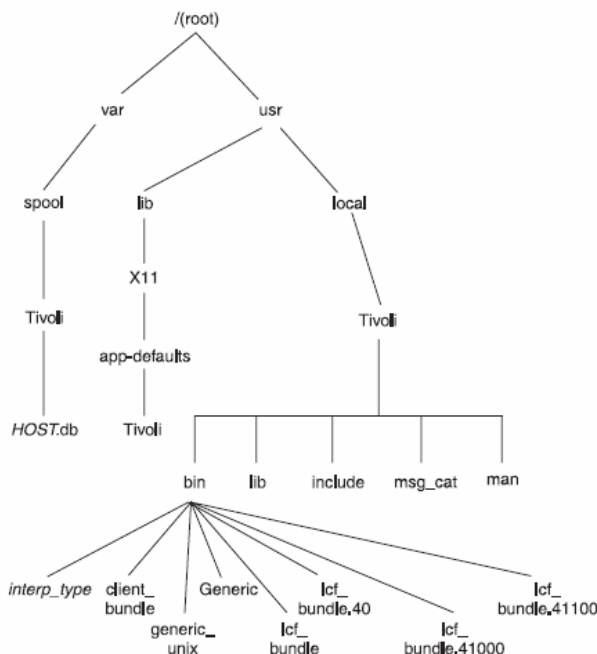
3.2. Inštalácia Tivoli Management Region serveru

Inštalácia začína vytvorením `/usr/local/Tivoli`, kde bude Tivoli Management Framework nainštalovaná. Nasledovne sa prepne do adresára `/usr/local/Tivoli`. Kde spustíme príkaz:

```
/cdrom/cdrom0/WPREINST.SH . /wserver -c /cdrom/cdrom0 \ BIN=/usr/local/Tivoli/bin  
\ LIB=/usr/local/Tivoli/lib \ ALIDB=/var/spool/Tivoli/teetmr00.db \ MAN=  
/usr/local/Tivoli/man \ APPD=/usr/lib/X11/app-defaults \  
CAT=/usr/local/Tivoli/msg_cat \ LK=1558698997 \ IP=manageIT \ AutoStart=1 \  
SetPort=1 \ CreatePaths=1
```

Tivoli Management Region Server bol nainštalovaný na linuxovom servery Red Hat Enterprise Linux 4 – hostname – teetmr00. TMR bol nainštalovaný do adresára `/usr/local/Tivoli`.

Celá štruktúra Tivoli Management Region vyzerá nasledovne:



Obr. 3.2: Adresárová štruktúra Tivoli Management Region

`/var/spool/Tivoli/teetmr00.db`

Obsahuje databázu súborov pre Tivoli servery alebo manažované uzly

`/usr/lib/X11/app-defaults`

Obsahuje zdrojové súbory pre X11.

`/usr/local/Tivoli/bin`

Obsahujú nasledovné Tivoli Enterprise binárnych adresárov:

`/client_bundle` – obsahuje súbory potrebné na zavedenie klientskej inštalácie, tento adresár je nainštalovaný len na Tivoli servery

`/generic` – obsahuje web stránky a súbory jazykov

/generic_unix – obsahuje generické binárne súbory pre všetky podporované platformy
/lcf_bundle – obsahuje binárne súbory pre endpointy vyžadované pre každú podporovanú platformu, na ktorej beží buď Tivoli Management Framework 3.6 alebo nižší

/lcf_bundle.40 - obsahuje binárne súbory pre endpointy vyžadované pre každú podporovanú platformu, na ktorej beží Tivoli Management Framework 3.7

/lcf_bundle.41000 - obsahuje binárne súbory pre endpointy vyžadované pre každú podporovanú platformu, na ktorej beží Tivoli Management Framework 4.1

/lcf_bundle.41100 - obsahuje binárne súbory pre endpointy vyžadované pre každú podporovanú platformu, na ktorej beží Tivoli Management Framework 4.1.1

/usr/local/Tivoli/lib

Obsahuje podadresár pre každú platformu, na ktorej je spustená jedna zo súčastí Tivoli infraštruktúry zaisťujúca chod monitoringu endpointov.

/usr/local/Tivoli/include

Obsahuje hlavičkové súbory používané Tivoli Application Development Environment

/usr/local/Tivoli/msg_cat

Obsahuje katalóg Tivoli správ

/usr/local/Tivoli/man

Obsahuje manuály pre Unix

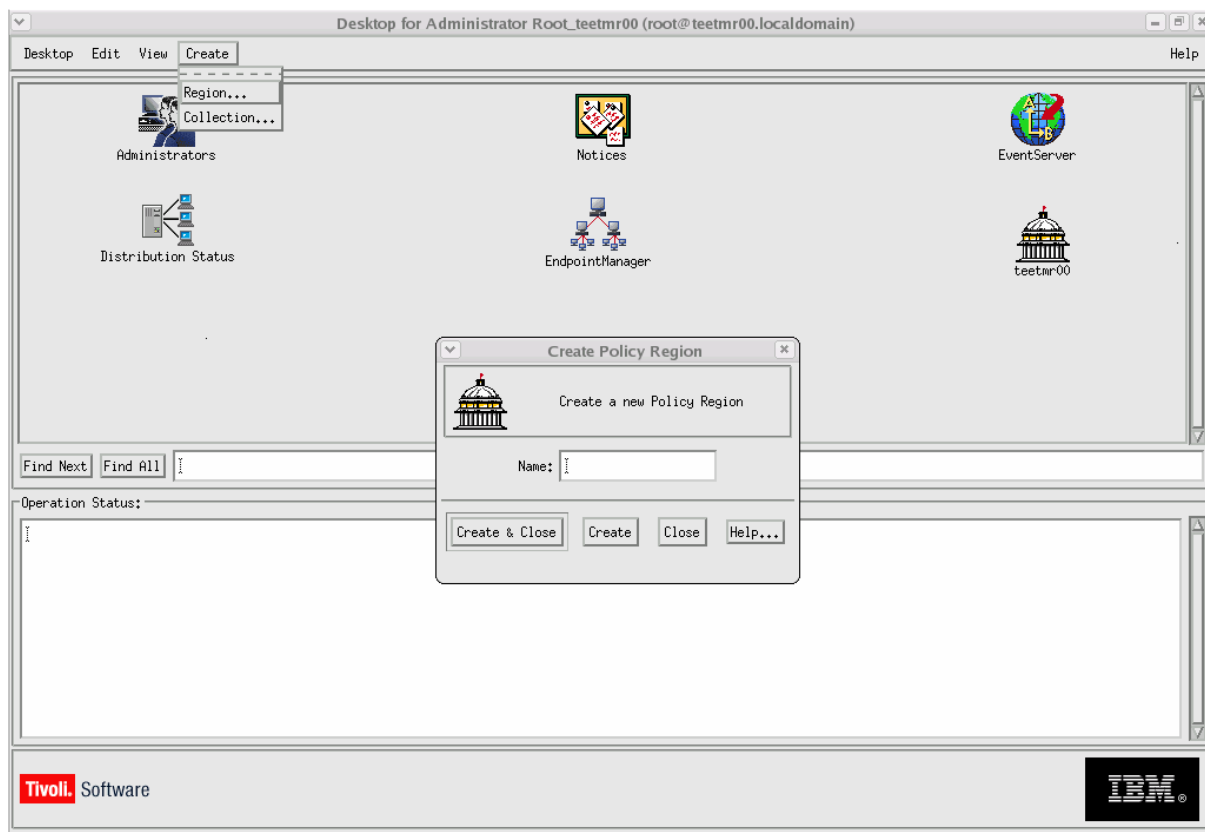
Týmto som popísal základnú štruktúru Tivoli Management Framework, ktorá bola nainštalovaná na teetmr00.

Na to aby sme mohli spravovať TMR musíme vytvoriť administrátorský účet nasledujúcim príkazom :

```
wcrtadmin -l Root_teetmr00@teetmr00.localdomain \ -r  
global,user:super:install_client \ -r global,install_product:backup:restore:senior \  
Installation-Manager
```

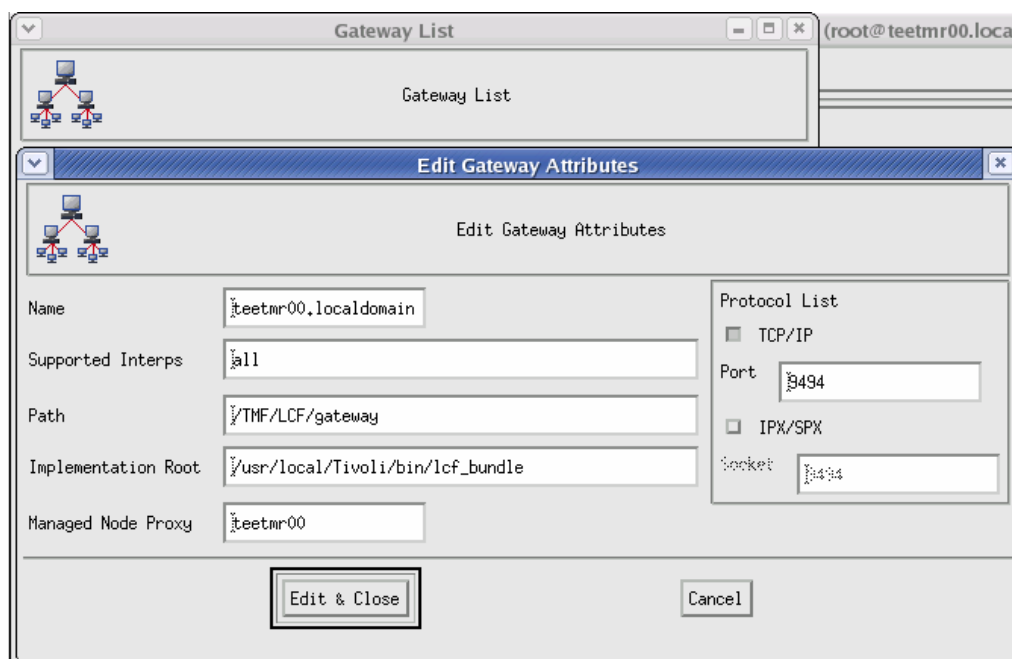
Po tomto kroku sa pomocou príkazu „tivoli“ dostávame do nainštalovaného produktu Tivoli Desktop, ktorý slúži k administrácii Tivoli regiónu .

Pomocou Tivoli Desktop vytvoríme Tivoli región „teetmr00“.



Obr. 3.3: Tivoli Desktop pre Linux

Na servery teetmr00 pomocou endpoint manažéra vytvoríme gateway *teetmr00.localdomain-tee-gaw*



Obr. 3.4: Vytvorenie Tivoli Gatewaye

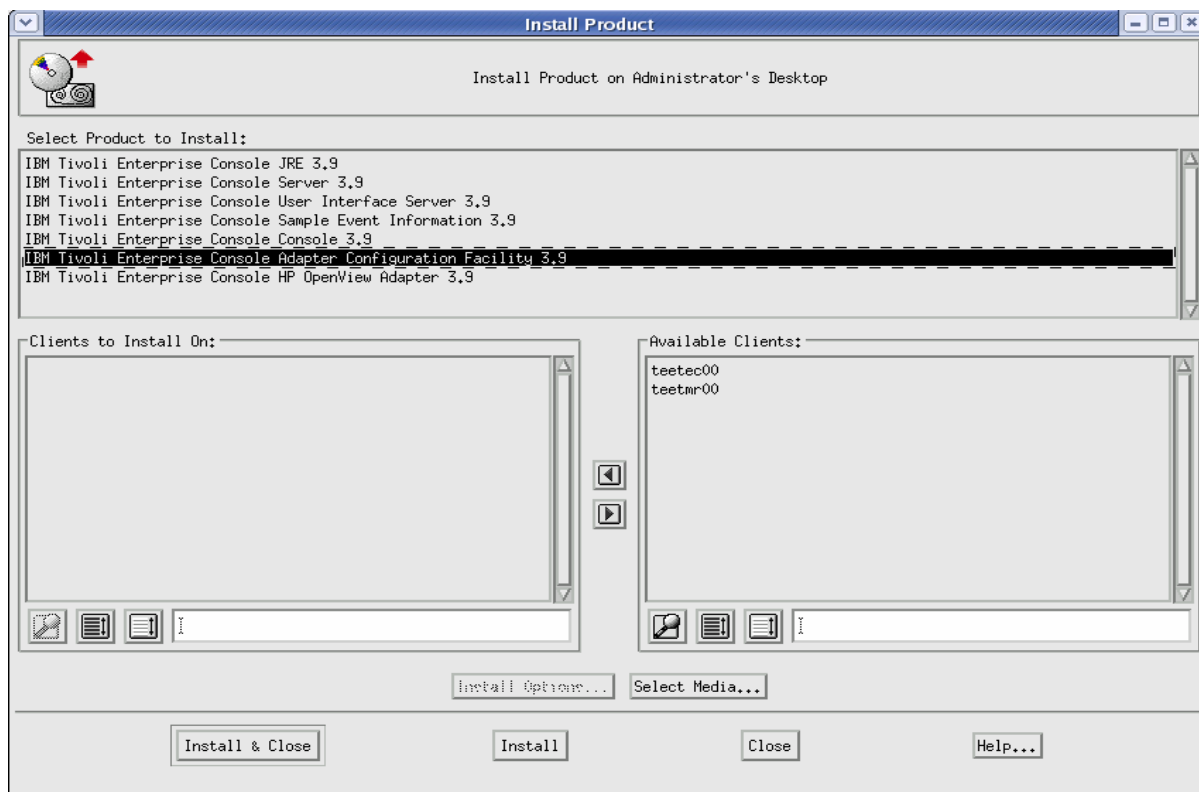
Keď máme takto predpripravený TMR server, môžeme prejsť k inštalácii Tivoli Enterprise Console serveru, ktorý bude taktiež nainštalovaný na Red Hat Enterprise Linux 4.

3.3. Inštalácia Tivoli Enterprise Console serveru:

Na linuxovom servery teetec00 najskôr nainštalujeme Java podporu z balíkov *Tivoli Java Client Framework 4.1.1*, *Java 1.3 for Tivoli* a *Tivoli Java RDBMS Interface Module JRIM 4.1.1*. Keď Java podpora nainštalovaná nainštalujeme balík *Tivoli Enterprise Console JRE 3.9*, následne na to nainštalujeme balík *Tivoli Enterprise Console Server 3.9*. Server reštartujeme (po opätovnom naboťovaní systému sa automaticky naštartuje na servery *teetec00* proces *oserv*, ktorý sa prihlási do TMR a bude naň možné doinštalovať Tivoli komponenty pomocou Tivoli Desktop) a prepne sa na server *teetrm00*. Ďalej ako som už zmienil bude inštalácia prebiehať pomocou Tivoli Desktop nainštalovanej na *teetmr00*. Na hlavnom paneli v záložke Desktop/Install vyberieme možnosť /Install Product . Na server *teetec00* postupne nainštalujeme:

IBM Tivoli Enterprise Console User Interface Server 3.9

IBM Tivoli Enterprise Adapter Configuration Facility 3.9

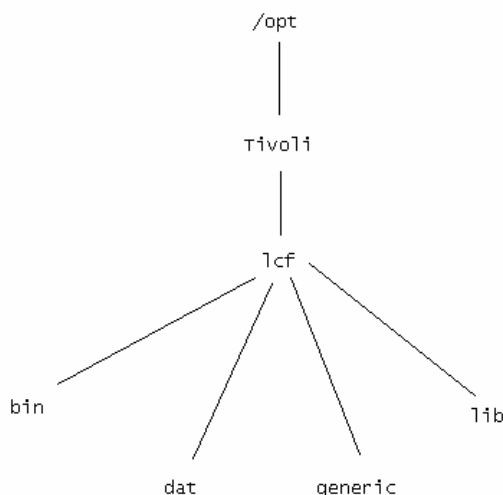


Obr. 3.5: Inštalácia Tivoli produktov na teetec00

Po nainštalovaní týchto Tivoli komponentov je nutné nainštalovať RDBMS server, ktorý v záujme rozdelenia vyťaženia jednotlivých Tivoli serverov nainštalujem na server *teetec00*. RDBMS server bude využívať databázu DB2 v8.1 od IBM. Na server boli nainštalované 2 databázy do adresárov */home/db2inst* a */home/db2mdist*. Po inštalácii a konfigurácii DB2 pre RDBMS je monitorovacia štruktúra predpripravená pre monitoring, jediné čo v tejto štruktúre chýba sú monitorované systémy.

3.4. Inštalácia Tivoli Endpointu na linuxový systém:

Tivoli endpoint nainštalujem z balíčku *tivoli_ep_linux-41146-2.i586.rpm*.
Tivoli endpoint je nainštalovaný do v nasledujúcej štruktúre:



Obr. 3.6: Adresárová štruktúra nainštalovaného endpointu na linuxový systém

/opt/Tivoli/lcf/bin

Obsahuje binárne súbory potrebné na chod endpointu.

/opt/Tivoli/lcf/dat

Obsahuje nastavenia endpoint (resource modely, záznamy o gateway, logy).

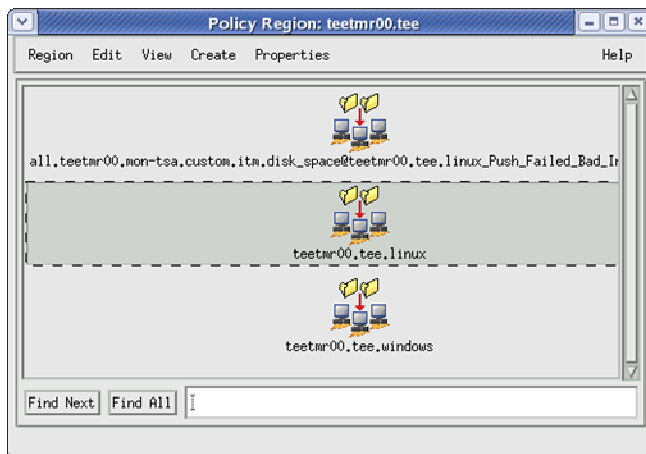
/opt/Tivoli/lcf/generic

Obsahuje súbory Tivoli správ a jazykov.

/opt/Tivoli/lcf/lib

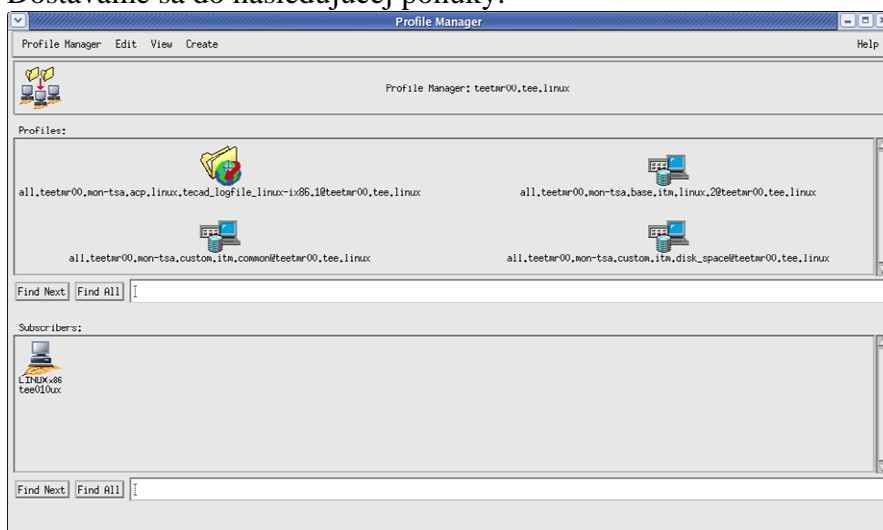
Obsahuje linuxové knižnice potrebné pre endpoint.

Po nainštalovaní Endpointu nie je potrebné na tomto servery už nič inštalovať. Avšak takto nainštalovaný endpoint ešte neposkytuje informácie o dianí na servery do Tivoli gateway, to čo bude odosielať sa distribuuje napríklad pomocou Tivoli Desktop. Na servery *teetmr00* zapneme Tivoli Desktop a otvoríme nami vytvorený región *teetmr00*. Nasledovne otvoríme pod región *teetmr00.tee* a v ňom región pre linuxové resource modely *teetmr00.tee.linux*.



Obr. 3.7: Policy regióny

Dostávame sa do nasledujúcej ponuky:



Obr. 3.8: Linux Policy región

Kde pomocou ponuky *Profile Manager* -> *Subscribers* podpíšeme náš endpoint tee010ux do tohto subregiónu pre linuxový operačný systém. Keď je tento endpoint v tomto subregióne podpísaný, môžeme naň distribuovať resource modely. Na tento endpoint nadistribuuujeme ponúkané resource modely, ktoré zaručujú monitoring voľného miesta na disku, stavu služieb, stav CPU, stav virtuálnej pamäte, dostupnosť systému a správy o reštartoch.

Pomocou príkazu *wmdist -l* zistíme identifikačné číslo operácie:

```
all.teetmr00.mon-tsa.base.itm.linux.2@teetmr00.tee.linux(install) 1558698997.24      1      1(100%)      1(100%)
```

Obr. 3.9: Posledný riadok vylistovaných distribučných operácií

Identifikačné číslo tejto operácie je 24 , takže pomocou príkazu *wmdist -e 24* zistíme stav distribúcie.

```
[root@teetmr00 TEC]# wmdist -e 24
```

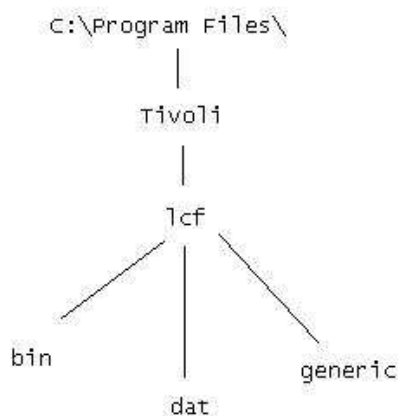
Name	Status	Start Time	End Time
teetmr00	SUCCESSFUL	2009.06.01 15:38:42	2009.06.01 15:42:01
tee010ux	SUCCESSFUL	2009.06.01 15:38:42	2009.06.01 15:42:01

Obr. 3.10: Záznam o úspešnej distribučnej operácii na endpoint tee010ux

Predošlý obrázok ukazuje na fakt, že resource modely boli úspešne odoslané z TMR cez Tivoli Gateway na endpoint tee010ux, ktorý resource modely úspešne prijal. Po tomto stačí reštartovať na servery tee010ux službu lcf a endpoint začne odosielať správy o systéme na gateway, ktorá ich prepošle na TMR a to do Event serveru na teetec00.

3.5. Inštalácia Endpointu a Tivoli Desktop na Microsoft Windows Server 2003 R2 Enterprise Edition:

Na nainštalovaný server s OS Windows Server 2003 nakopírujeme inštalačné súbory pre Tivoli Desktop a Tivoli Endpoint. Najskôr nainštalujeme Tivoli Endpoint do C:\Program Files\Tivoli\lcf. Štruktúra vyzerá nasledovne:



Obr. 3.11: Adresárová štruktúra Tivoli endpointu na OS Windows

C:\Program Files\Tivoli\lcf\bin

Obsahuje súbory potrebné na chod endpointu.

C:\Program Files\Tivoli\lcf\dat

Obsahuje nastavenia endpoint (resource modely, záznamy o gateway, logy).

C:\Program Files\Tivoli\lcf\generic

Obsahuje súbory Tivoli správ a jazykov.

Po nainštalovaní overíme pomocou services.msc či sa nainštalovala služba lcf a či je nastavená na automatické spustenie. Na servery teetmr00 zapneme Tivoli Desktop a otvoríme nami vytvorený región *teetmr00*. Nasledovne otvoríme subregión *teetmr00.tee* a v ňom región pre windowsové resource modely *teetmr00.tee.windows*, kde pomocou ponuky *Profile Manager->Subscribers* podpíšeme náš endpoint 2003ep do tohto subregiónu pre windowsový operačný systém. Keď je tento endpoint v tomto subregióne podpísaný, môžeme naň distribuovať resource modely. Na tento endpoint nadistribuuujeme ponúkané resource modely, ktoré zaručujú monitoring voľného miesta na disku, stavu služieb, stav CPU, stav virtuálnej pamäte, dostupnosť systému a správy o reštartoch.

Pomocou príkazu *wmdist -l* zistíme identifikačné číslo operácie. Identifikačné číslo tejto operácie je 28, takže pomocou príkazu *wmdist -e 28* zistíme stav distribúcie.

```
[root@teetmr00 TEC]# wmdist -e 28
```

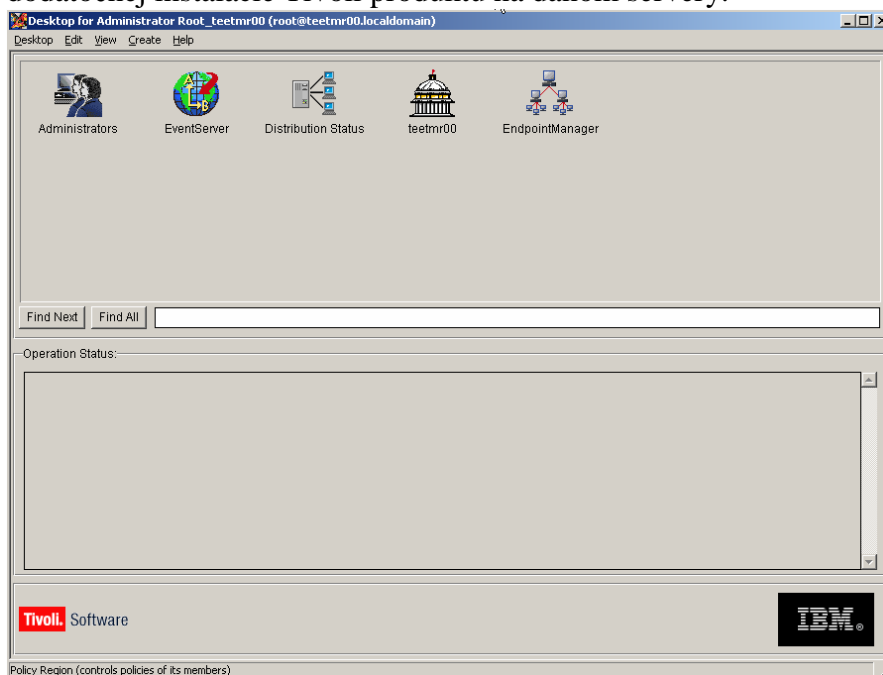
Name	Status	Start Time	End Time
teetmr00	SUCCESSFUL	2009.06.01 16:03:09	2009.06.01 16:15:34
2003ep	SUCCESSFUL	2009.06.01 16:09:04	2009.06.01 16:15:34

Obr. 3.12: Záznam o úspešnej distribučnej operácii na endpoint 2003ep

Predošlý obrázok ukazuje na fakt, že resource modely boli úspešne odoslané z TMR cez Tivoli Gateway na endpoint *2003ep*, ktorý resource modely úspešne prijal. Po tomto stačí reštartovať na servery *2003ep* službu *lcf*d a endpoint začne odosielať správy o systéme na gateway, ktorá ich prepošle na TMR a to do Event serveru na *teetec00*.

Tivoli Desktop sa nainštaluje do *C:\Program Files\Tivoli\Desktop*.

Táto inštalácia nie je nutná, Tivoli Desktop pre Windows nám len umožňuje spravovať Tivoli Management Framework z akéhokoľvek serveru, na ktorom je Tivoli Desktop nainštalovaný. Týmto sme získali to isté prostredie na OS Windows bez akejkoľvek dodatočnej inštalácie Tivoli produktu na danom servery.



Obr. 3.13: Tivoli Desktop pre Windows

Keď máme takto nainštalovanú štruktúru TMR, TEC a Endpointov môžeme ju spustiť.

1, Servery *2003ep* a *tee010ux* môžeme zapnúť. Endpointy týchto serverov sa prihlásia na gateway hneď ako zistia že gateway je dostupná.

2, Z monitorovacej infraštruktúry je potreba najskôr zapnúť server *teetmr00* na ktorom sa automaticky spustí Tivoli Management Framework.

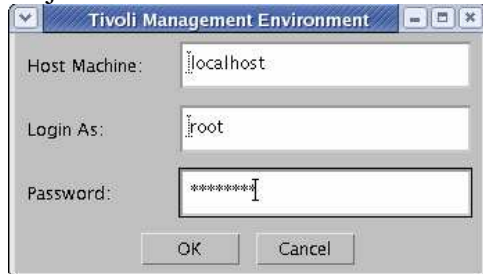
3, Popri tom ako nám nabieha *teetmr00* môžeme spustiť server *teetec00*. Na tomto servery sa spustí proces *oserv*. Po prihlásení s účtom *root* a heslom *passwd* sa prepne do adresára */usr/local/Tivoli/repos/install*, kde sa prepne pomocou príkazu *su - db2mdist* na užívateľa *db2mdist* a spustíme príkazom *db2start* databázu *mdist2*. funkčnosť databázy overíme príkazom *wrimtest -l mdist2*. Následne sa prepne na užívateľa *db2inst* a zapneme databázu príkazom *db2start*, jej dostupnosť overíme príkazom *wrimtest -l tec*. Ak prebehne všetko v poriadku ako v mojom prípade prepne sa na *teetmr00* do Tivoli Desktop a pravým tlačítkom klikneme na ikonu Event Serveru a vyberieme z ponuky *Start-up*, čo v ideálnom prípade zapne Event server a načíta databázy. Po tomto sa znova prepne na *teetec00*, kde do terminálu

zadáme postupne tieto príkazy, ktoré naštartujú Tivoli prostredie Pre Tivoli Enterprise konzolu:

```
ls -la /etc/Tivoli/setup_env.sh  
./etc/Tivoli/setup_env.sh
```

4, Potom zadáme príkaz *tec_console*, čím spustíme samotnú monitorovaciu konzolu:

Objaví sa nasledovné okno:



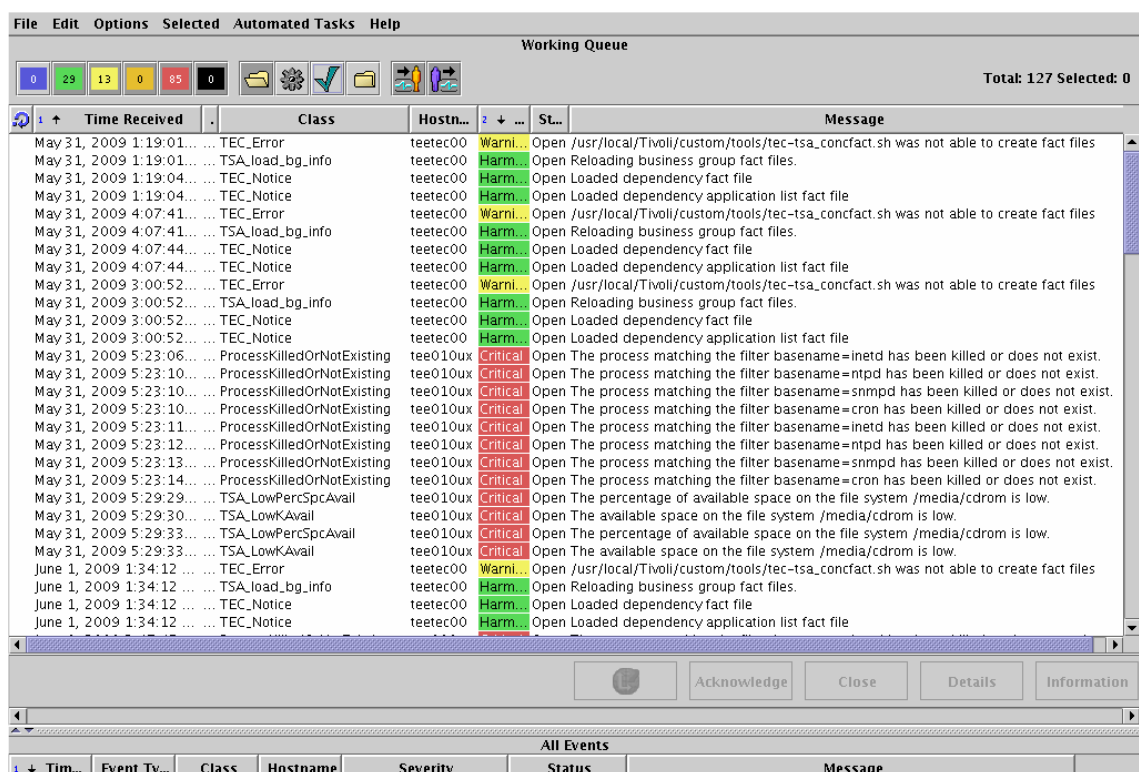
Obr. 3.14: Prihlasovacie okno do TECu

Host Machine: *192.168.1.2* alebo *localhost*

Login As: *root*

Password: *passw0rd*

Po zadaní uvedených hodnôt sa dostávame do užívateľského rozhrania Tivoli Enterprise Console.



Obr. 3.15: Tivoli Enterprise konzola pre región teetmr00

Keď máme spustenú konzolu, môžeme začať vykonávať operácie na monitorovaných serveroch.

Na server *2003ep* vypneme niektoré služby, napríklad Windows time, Print Spooler a Windows automatic updates.

Po tom ako boli tieto služby vypnuté do monitorovacej konzoly boli dotučené nasledujúce hlášky:

```
June 1, 2009 7:17:43 PM... TSA_WinServiceNotRunning Critical 2003ep Open The Service Print Spooler (Spooler) is not running
June 1, 2009 7:17:45 PM... TSA_WinServiceNotRunning Critical 2003ep Open The Service Windows Time (W32Time) is not running
June 1, 2009 7:17:45 PM... TSA_WinServiceNotRunning Critical 2003ep Open The Service Automatic Updates (wuauserv) is not running
```

Obr. 3.16: Hlášky z endpointu 2003ep po vypnutí služieb

Operátor v tomto prípade na servery dané služby naštartuje na servery.

Pre ďalšiu ukážku reštartneme server *tee010ux*, v priebehu pár minút prídu do konzoly nasledujúce hlášky:

```
June 1, 2009 8:25:35 PM... TSA_reboot Warning tee010ux Open The host has rebooted
June 1, 2009 8:25:34 PM... ProcessKilledOrNotExist Critical tee010ux Open The process matching the filter basename=cron has been killed or does not exist.
June 1, 2009 8:25:33 PM... ProcessKilledOrNotExist Critical tee010ux Open The process matching the filter basename=snmpd has been killed or does not exist.
June 1, 2009 8:25:32 PM... ProcessKilledOrNotExist Critical tee010ux Open The process matching the filter basename=ntpd has been killed or does not exist.
June 1, 2009 8:25:31 PM... ProcessKilledOrNotExist Critical tee010ux Open The process matching the filter basename=inetd has been killed or does not exist.
June 1, 2009 8:25:27 PM... ProcessKilledOrNotExist Critical tee010ux Open The process matching the filter basename=inetd has been killed or does not exist.
June 1, 2009 8:25:27 PM... ProcessKilledOrNotExist Critical tee010ux Open The process matching the filter basename=ntpd has been killed or does not exist.
June 1, 2009 8:25:27 PM... ProcessKilledOrNotExist Critical tee010ux Open The process matching the filter basename=snmpd has been killed or does not exist.
June 1, 2009 8:25:27 PM... ProcessKilledOrNotExist Critical tee010ux Open The process matching the filter basename=cron has been killed or does not exist.
```

Obr. 3.17: Hlášky z endpointu teetec00 po reštartovaní serveru

Z týchto hlášiek sa dá vyčítať že na servery boli násilne vypnuté služby a server bol následne reštartovaný. Daný resource model *TSA_reboot* je užitočný hlavne v situáciách, kedy na servery beží určitá aplikácia, ktorá vyžaduje manuálny štart alebo je potrebná iná procedúra po reštarte hlavne u serverov, ktoré riadia výrobné linky a tak ďalej.

Nasledujúca hláška upozorňuje na málo miesta na disku, jedná sa síce o CD–rom mechaniku, ale táto hláška dokazuje, že Resource model *TSA_free_space* funguje tiež správne pretože nám hlási, že na médiu /cdrom je nízka, čo je pravda keďže na tomto médiu nie je žiadne voľné miesto.

```
June 1, 2009 6:07:09 PM... TSA_LowPercSpcAvail Critical tee010ux Open The percentage of available space on the file system /media/cdrom is low.
```

Obr. 3.18: Hláška z endpointu tee010ux o nedostatočnom mieste na disku

Ako ďalšiu testovaciu úlohu som vybral vypnutie serveru *2003ep*, konzola reagovala hláškou:

```
June 2, 2009 12:42:13 A... HeartBeat_EndpointUnreach... Warning 2003ep Open Tivoli lcf endpoint is unreachable.
```

Obr. 3.19: Hláška o nedostupnosti serveru 2003ep

Po opätovnom naštartovaní serveru *2003ep* konzola zahlásila, že monitoring na servery *2003ep* je naštartovaný a je v poriadku:

```
June 2, 2009 12:49:05... HeartBeat_DMAgentAlive 2003ep Harmless Open The IBM Tivoli Monitoring engine is alive.
```

Obr. 3.20: Hláška o dostupnosti serveru 2003ep

Do konzoly prichádzali hlášky aj z TEC serveru, ktoré vyzerali nasledovne:

```
June 1, 2009 7:34:30... TEC_Notice teetec00 Harmless Open Loaded dependency fact file
June 1, 2009 7:34:30... TEC_Notice teetec00 Harmless Open Loaded dependency application list fact file
June 1, 2009 7:34:29... TEC_Error teetec00 Warning Open /usr/local/Tivoli/custom/tools/tec-tsa_contact.sh was not able to create fact files
```

Obr. 3.21: Informačné hlášky z TEC serveru

Tieto hlášky signalizovali to, že Event Server nemal dosť prostriedkov na to, aby uložil určité dáta a konzola musela byť reštartovaná. Tento problém bol spôsobený nedostatočným množstvom prostriedkov, pretože na som mal k dispozícii počítač so 4 GB RAM a 2 jadrovým procesorom, čo je aj na takú minimalizovanú infraštruktúru málo. Tento problém, nebol až taký nepríjemný pretože simuloval problémy, ktoré

občas nastávajú s Tivoli Management Framework. Medzi najčastejšie problémy patrí nedostupnosť Tivoli gateway, nedostupnosť Tivoli endpointu alebo nekorektné bežanie resource modelov.

Na ukážku popíšem základné príkazy používané v Tivoli a ich výstupy v tejto infraštruktúre:

tivoli – na hlavnom TMR servery spustí Tivoli Desktop

tec_console – spustí Tivoli Enterprise konzolu

wep ls - vylisuje všetky manažované uzly

```
[root@teetmr00 ~]# wep ls
G      1558698997.1.615  teetmr00.localdomain-tee-gaw
1558698997.10.522+#TMF_Endpoint::Endpoint# 2003ep
1558698997.5.522+#TMF_Endpoint::Endpoint# tee010ux
```

Obr. 3.22: výstup wep ls

wgateway - vylisuje všetky Tivoli brány

```
[root@teetmr00 ~]# wgateway
Object                                Name                                Status
1558698997.1.615                    teetmr00.localdomain-tee-gaw u
```

Obr. 3.23: výstup wgateway

Pri použití wgateway <gateway> start/stop – naštartuje/zastaví danú gateway

wep <endpoint> status - vypíše aktuálny stav endpointu/gatewaye

```
[root@teetmr00 ~]# wep 2003ep status
2003ep is alive.
```

Obr. 3.24: výstup wep 2003ep status

wep <enspoint> - vypíše informácie, ktoré poskytol endpoint pri poslednom prihlásení na gateway

```
[root@teetmr00 ~]# wep 2003ep
      object 1558698997.10.522+#TMF_Endpoint::Endpoint#
      label 2003ep
      version 41146
      id DFCA383A-4D33-11DE-B396-81CA772595ED
      gateway 1558698997.1.615#TMF_Gateway::Gateway#
      pref_gateway 1558698997.1.615#TMF_Gateway::Gateway#
      netload OBJECT_NIL
      interp w32-ix86
      login_mode desktop|constant
      protocol TCPIP
      address 192.168.1.104+9495
      mac address (WOL) NONE
      subnet mask (WOL) NONE
      policy OBJECT_NIL
      httpd tivoli:@DQ/+)[P
      alias OBJECT_NIL
      crypt_mode NONE
      upgrade_mode enable
      ep_status connected
      status_time 2009/06/02-00:26:21
      error_codes
      last_login_time 2009/06/02-00:22:27 [Passed]
      last_logout_time 2009/06/02-00:12:23 [Passed]
      last_migration_time 2009/06/02-00:19:54 [Passed]
      last_downcall_time 2009/06/02-00:25:05 [Passed]
      last_upcall_time 2009/06/02-00:23:34 [Passed]
      last_ep_check_time 2009/06/01-23:45:59 [Failed]
      allow_distribution_control off
      hostname 2003ep
      hostbyname 2003ep
```

Obr. 3.25: výstup wep 2003ep

wdmlseng -e <endpoint> - vypíše stav resource modelov nainštalovaných na endpointe

```
[root@teetmr00 ~]# wdmlseng -e tee010ux
```

Forwarding the request to the endpoint:

```
tee010ux 1558698997.5.522+#TMF_Endpoint::Endpoint#
```

The following profiles are running:

```
all.teetmr00.mon-tsa.base.itm.linux.2#teetmr00
```

```
TSA_Ping: Running
```

```
DMXCpu: Running
```

```
DMXMemory: Running
```

```
TSA_DMXFileSystem: Running
```

```
DMXNetworkInterface: Running
```

```
TSA_reboot: Running
```

```
DMXSpp: Running
```

```
DMXProcess: Running
```

Obr. 3.26: výstup *wdmlseng -e tee010ux*

odadmin odlist – vypíše všetky Manage nody v infraštruktúre

```
[root@teetmr00 ~]# odadmin odlist
```

Region	Disp	Flags	Port	IPAddr	Hostname(s)
1558698997	1	ct-	94		teetmr00.localdomain, teetmr00
-	2	ct-	94		teetec00.localdomain, teetec00

Obr. 3.27: výstup *odadmin odlist*

4. Záver

V mojej bakalárskej práci som dostal za úlohu vytvoriť monitorovaciu infraštruktúru pre serverové systémy. Nakoľko moje počiatočné odhady počítali s 10 virtuálnymi servermi, som bol nútený obmedziť počet serverov len na 4 pretože by to môj počítač nezvládol. Avšak aj pri menšom množstve serverov sa dá naznačiť ako Tivoli prostredie funguje. V mojej práci som bol schopný nainštalovať kompletne monitorovacie prostredie Tivoli rozdelené na 2 linuxové servery z dôvodu ich rovnomerného vytťaženia a 2 monitorované servery z toho jeden na linuxovej a druhý na platforme Windows. Inštalácia prebiehala postupne ako je popísaná v mojej práci. Pred samostatnou inštaláciou je nutné si dopredu rozvrhnúť aké komponenty budeme v našej infraštruktúre potrebovať a ako by mali byť postavené. Pre Tivoli management Region server a Tivoli Enterprise Console server sa odporúčajú servery bežiacie na unixovom OS pretože sú stabilnejšie a Tivoli prostredie bolo vyvíjané práve na tomto operačnom systéme. Ďalšia hlavná podmienka pre tieto 2 servery je dostatočná kapacita sieťového pripojenia. V záverečnej časti práce je uskutočnené meranie, čím bola overená funkčnosť nainštalovanej Tivoli infraštruktúry. Endpointy svoju úlohu splnil tým, že na nich nainštalované resource modely odosieli pravdivé informácie, ktoré sa dostali do konzoly až k operátorovi. Monitoring serverov a akéhokoľvek iného prvku infraštruktúry slúži nielen na zistenie problému v reálnom čase ale hlavne slúži na zabezpečenie dostupnosti prvkov ktoré zákazník využíva. Takto je monitoring schopný odhaliť nefunkčný disk v RAID-ovom poli, čo zabezpečí pri dostatočne rýchlo vymenení poškodeného disku udržanie dát. Pomocou takejto infraštruktúry je možné monitorovať aplikácie, ktoré operujú s výrobnými linkami a tak zabezpečiť minimálny výpadok výroby a tým aj straty na zákazníkovej strate.

5. Zoznam použitých zdrojov

1, IBM. *Tivoli Enterprise : Installation Guide*. 4th edition. [2002]. 416 s. ISBN GC32-0804-01.

2, IBM. *An Introduction to Tivoli Enterprise*. 1st edition., 1999. 906 s. 1. Dostupný z WWW: <www.redbooks.ibm.com>. ISBN SG24-5494-00.

3, IBM. *Tivoli Enterprise Console*. 1st edition., 2003. 67 s. 1. ISBN SC32-1238-00.

4, IBM. *Common Criteria Certification: Installing IBM DB2 Version 9.1 : Enterprise Server Edition for Linux, UNIX, and Windows– Revision 04.*, 2006. 278 s. 4. ISBN SC10-4315-00.

6. Zoznam použitých skratiek

TMR - Tivoli Management fFRAMEWORK

TEC – Tivoli Enterprise Console

RDBMS – Relational Database Management System

RIM – RDBMS Interface Module

TME – Tivoli Managed Enviroment

GUI - Graphical user interface

7. Prílohy

Celá infraštruktúra je rozdelená a napálená na 4 DVD

DVD1 – TMR.001

DVD2 – TEC.001

DVD3 – 2003ep.rar, tee010ux.rar, TMR.002, TMR.CRC

DVD4 - TEC.002, TEC.CRC, manual_start_infra.txt, 78513_Jankular_BC.pdf,